WHAT IS CLAIMED

1. A router for routing packets in a network, the router comprising:

a plurality of processing components configured to determine destination information for the packets, one of the processing components being an active processing component and the other of the plurality of processing components being non-active processing components within the router; and

SubA'

a plurality of routing engines configured to maintain routing tables that contain packet routing information and supply the routing tables to the processing components, one of the plurality of routing engines being an active routing engine and the other of the plurality of routing engines being non-active routing engines, at least one of the non-active routing engines receiving information from the active routing engine indicating whether the active routing engine is functioning, the at least one of the non-active routing engines being configured to assert itself as the active routing engine when the non-active routing engine fails to receive the information from the active routing engine.

2. The router of claim 1, further comprising:

a redundancy controller connected to the plurality of routing engines, the redundancy controller resetting the active routing engine when the non-active routing engine asserts itself as the active routing engine.

3. The router of claim 2, further comprising:

a redundancy switch connecting the redundancy controller and the plurality of routing engines to one of the plurality of processing components, the connected one of the plurality of processing components being selected based on a signal received from the redundancy controller.

4. The router of claim 3, further comprising:

a packet manager connected to the redundancy switch and configured to handle input/output functions related to incoming and outgoing packets.

5. The router of claim 3, wherein the plurality of processing components includes a first processing component and a second processing component and wherein the plurality of routing engines includes a first routing engine and a second routing engine.

A 6. The router of claim 5, wherein the redundancy controller includes:

a first servant circuit connected to and controlled by the first routing engine; and

a second servant circuit connected to and controlled by the second routing engine; wherein

the first servant circuit issues a reset command to the second servant circuit and the second routing engine based on a command from the first routing engine, and the second servant circuit issues the reset command to the first servant circuit and the first routing engine based on a command from the second routing engine, the reset

command causing the servant circuit receiving the reset command to relinquish control of the router.

7. The router of claim 6, wherein the redundancy controller further includes:

a plurality of switches, each of the plurality of switches receiving a first input from the first servant circuit and a second input from the second servant circuit, and outputting one of the first and second inputs.

8. The router of claim 7, wherein the plurality of switches are each organized as a logical AND gate.

9. The router of claim 7, wherein the first and second servant circuits and the plurality of switches implement deadlock recovery of the routing engines.

10. A router comprising:

a first routing engine;

a second routing engine; and

a redundancy controller circuit connected to the first and second routing engines and configured to reset one of the first and second routing engines and to allow the other of the first and second routing engines to become an active routing engine.

11. The router of claim 10, wherein the reset one of the routing engines enters a standby mode of operation.

12. The router of claim 10, further comprising:

a first processing component;

a second processing component; and

a redundancy switch connected to the redundancy controller circuit, the redundancy switch selectively connecting one of the first and second processing components to the redundancy controller circuit based on a signal received from the redundancy controller circuit.

13. The router of claim 12, wherein the redundancy controller circuit further comprises:

a first servant circuit connected to and controlled by the first routing engine; and

a second servant circuit connected to and controlled by the second routing engine; wherein

the first servant circuit issues a reset command to the second servant circuit and the second routing engine based on a command from the first routing engine, and the second servant circuit issues a reset command to the first servant circuit and the first routing engine based on a command from the second routing engine, the reset command causing the servant circuit receiving the reset command to relinquish control of the router.

14. The router of claim 13, wherein the redundancy controller further comprises:

a plurality of switches, each of the plurality of switches receiving a first input from the first servant circuit and a second input from the second servant circuit, and outputting one of the first and second inputs.

15. The router of claim 14, wherein the plurality of switches are each organized as a logical AND gate.

16. The router of claim 14, wherein the plurality of switches includes:

a first of the plurality of switches having an output connecting to the first processing component;

a second of the plurality of switches having an output connecting to the second processing component; and

a third of the plurality of switches having an output connected to the redundant switch.

17. A method of controlling a router having redundant components, including at least first and second routing engines coupled to a packet forwarding engine, the method comprising:

powering-up the first and the second routing engines in a standby state;

negotiating between the first and second routing engines which of the first and second routing engines is to be in an active state and which of the first and second routing engines is to remain in the standby state;

resetting, by the routing engine that is to be in the active state, control paths within the router such that the active routing engine communicates with the packet forwarding engine.

18. The method of claim 17, further comprising:

determining, by the active routing engine, whether first and second processing components are present in the router;

determining, by the active routing engine, whether the first and the second processing components are operating in the router;

determining, by the active routing engine, a preferred state for operating the first and second processing components;

selecting one of the first and second processing components based on the determinations of whether the first and second processing components are present in the router, whether the first and the second processing components are operating in the router, and the preferred operating state of the first and second processing components; and

activating the selected one of the first and second processing components.

19. The method of claim 17, further comprising:

receiving information at the non-active routing engine that indicates whether the active routing engine is still functioning properly; and

resetting the active routing engine and the control paths within the router such that the non-active routing engine becomes the active routing engine when the received information indicates that the active routing engine is no longer functioning properly.

20. A method of controlling a router having redundant components, including at least first and second routing engines coupled to a packet forwarding engine, the method comprising:

setting the first routing engine to an active state, the first routing engine communicating with the packet forwarding engine while in the active state;

setting the second routing engine to a standby state, the second routing engine, when in the standby state, monitoring the first routing engine for a failure in the first routing engine; and

the second routing engine assuming the active state when the second routing engine detects a failure in the first routing engine.

21. The method of claim 20, wherein the second routing engine assumes the active state by causing a servant circuit associated with the second routing engine to reset a servant circuit associated with the first routing engine, each of the serving circuits being independent of the respective routing engines. servants ?

22. The method of claim 20, wherein the second routing engine selects one of multiple redundant processing components in the packet forwarding engine when assuming the active state.

23. The method of claim 22, wherein the second routing engine communicates with the selected processing component.